

# FUNDAMENTAL OF PROGRAMMING IN C#

## INTRODUCTION TO PROGRAMMING

Yunghans Irawan

yirawan@nus.edu.sg

# Agenda

- What is Programming?
- Software Development Lifecycle
- Different kind of software
- Programming Languages
- Algorithms
- Explore programming with Blockly Games

# What is Programming

Making computers do what you want them to do  
by expressing your intentions and instructions  
clearly in a structured and un-ambiguous way



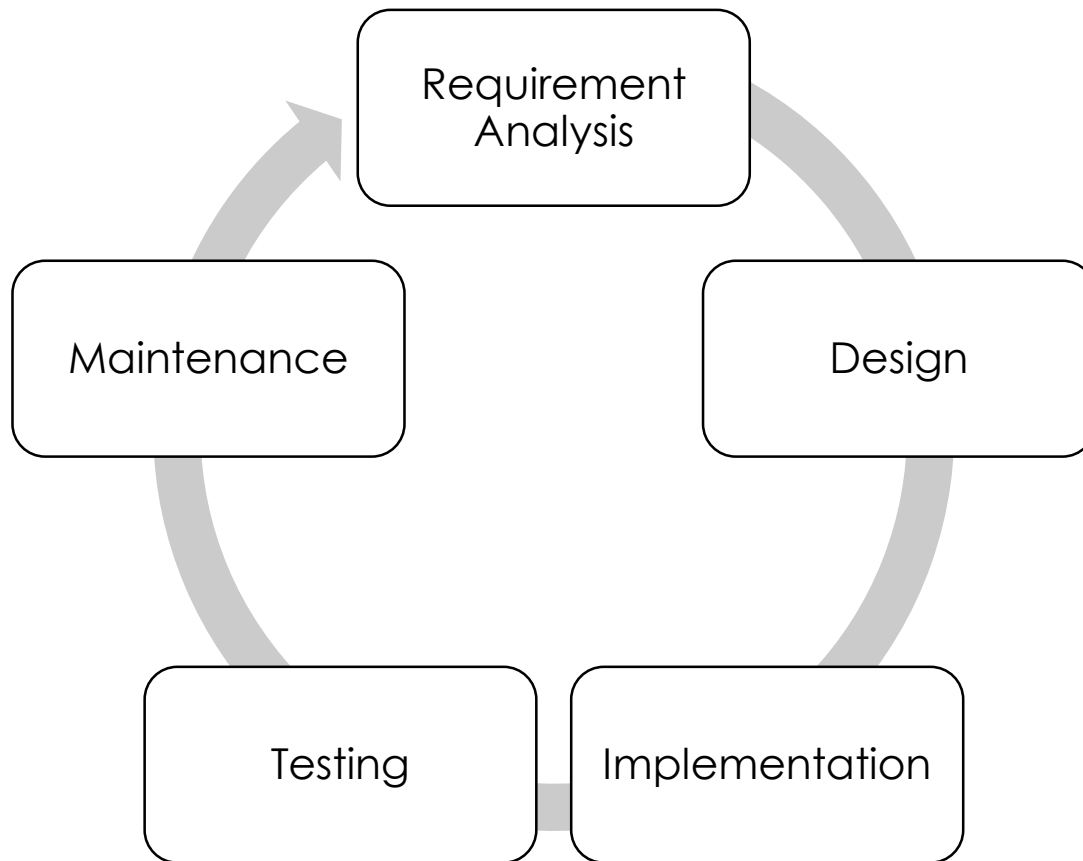
*programming language*

# Application of Programming

- Many different domains and industry
  - Websites
  - Apps
  - Games
  - Robots
  - Self-driving cars
  - Infrastructures
  - Healthcare
  - Entertainment
  - Arts
  - Movies special effects
  - And many more...

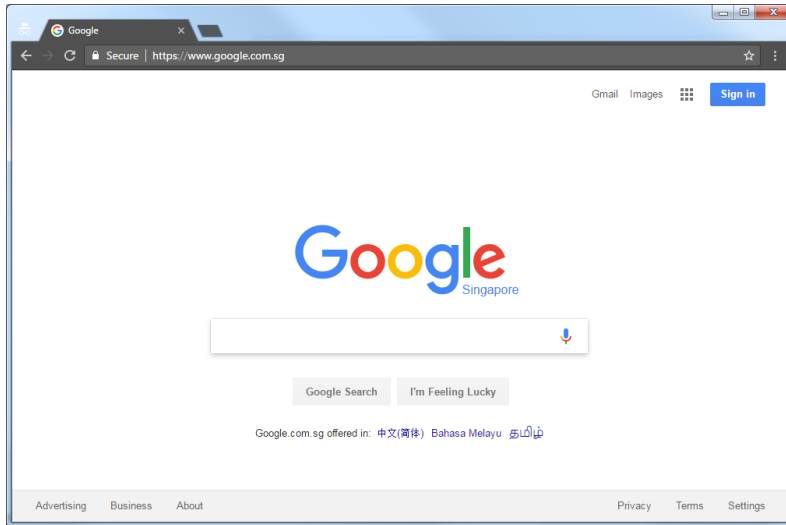
# Software Development Life Cycle

How people build software?

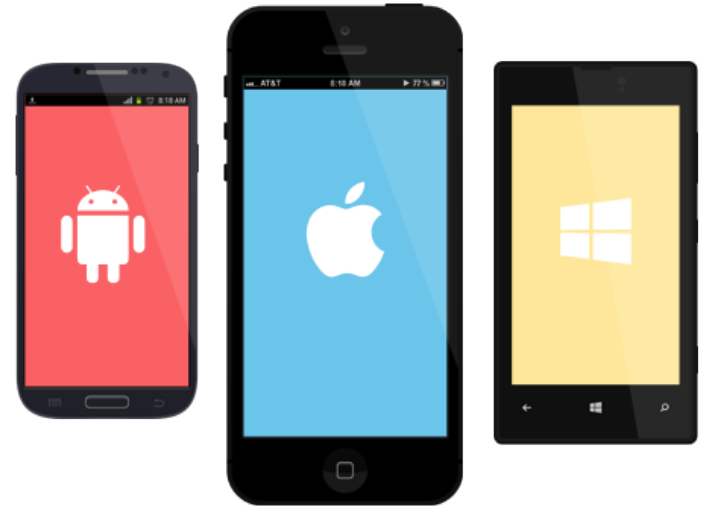


At which stage would programming skill be needed?

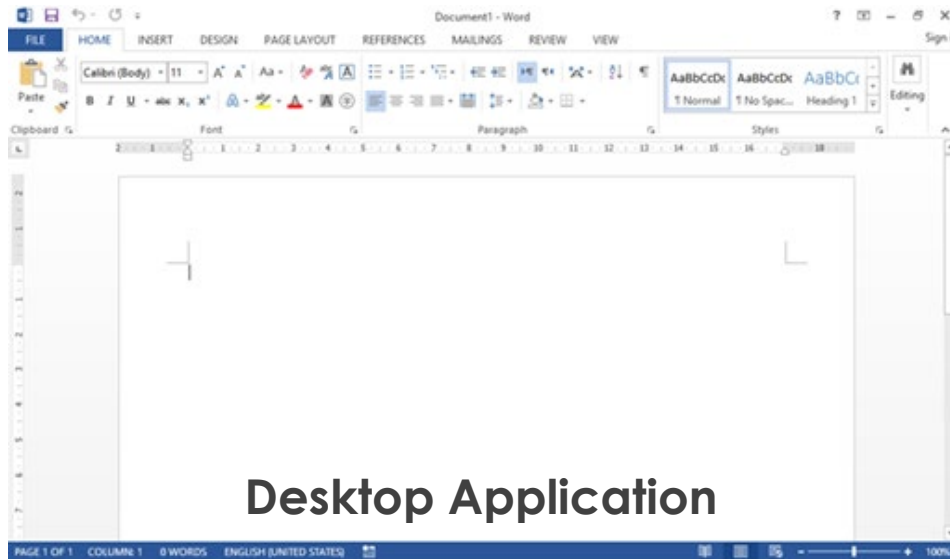
# Different Kinds of Software



**Web Application**



**Mobile Application**



**Desktop Application**



**Embedded Application**

# Software, Application and Program

- Software
  - A collection (or a package) of data and computer instructions (or programs) that tell computer how to work
  - Can contains computer programs, libraries, and data (such as documentation and digital media)
  - Software vs. hardware: digital vs physical assets
  - There are many different kinds of software e.g.: application, operating systems, etc.
- Application (or application software)
  - Software designed for users which contains a group of coordinated functions, tasks or activities

# Software, Application and Program

- System Software
  - Software that directly operates the hardware
  - Operating System: provide common services to run other softwares. E.g. Microsoft Windows, Linux, MacOS, Android, iOS
  - Device Drivers: software that allows operating system to control a particular type of device
- Program
  - A collection of instructions that performs a specific task when executed by a computer
  - Written by computer programmers using a programming language and stored as source code file(s).
  - A program may be compiled by a compiler into machine code that can be executed directly by machine (e.g. C language), or
  - A program may be executed directly from the source code by an interpreter (e.g. Python, JavaScript)



# Software, Application and Program

- Library
  - A packaged collection of behaviour or capabilities that can be called (or invoked) by a program.
  - Libraries allows programs to be written more productively because common behaviour and functionality can be packaged into libraries and made available for the programmers.
  - Programming languages have standard libraries that is shipped by default with the programming tools.

# Programming Language

- A language system used to instruct computers
  - Each language has their own syntax and rules

4GL Language
High Level Language
Low Level Language
Assembler Language
Machine Language

Higher level = close to how human  
express themselves



Lower level = closer to the  
technical implementation of the  
computer

# Programming Language

## Assembly Language (Low Level)

```
mov ax, 0x000D    ; set 320x200 graphics mode
int 0x10          ; bios video services
mov ax, 0x0C07    ; put pixel in white color
xor bx, bx        ; page number 0
mov cx, 200       ; x position 200
mov dx, 100       ; y position 200
l1: int 0x10       ; bios video services
dec dx            ; decrease y position
loop l1           ; decrease x position and repeat
mov ah, 0         ; service 0 - get keystroke
int 0x16          ; bios keyboard services
mov ax, 0x0003    ; 80x25 text mode
int 0x10          ; bios video services
mov ax, 0x4c00    ; terminate program
int 0x21
```

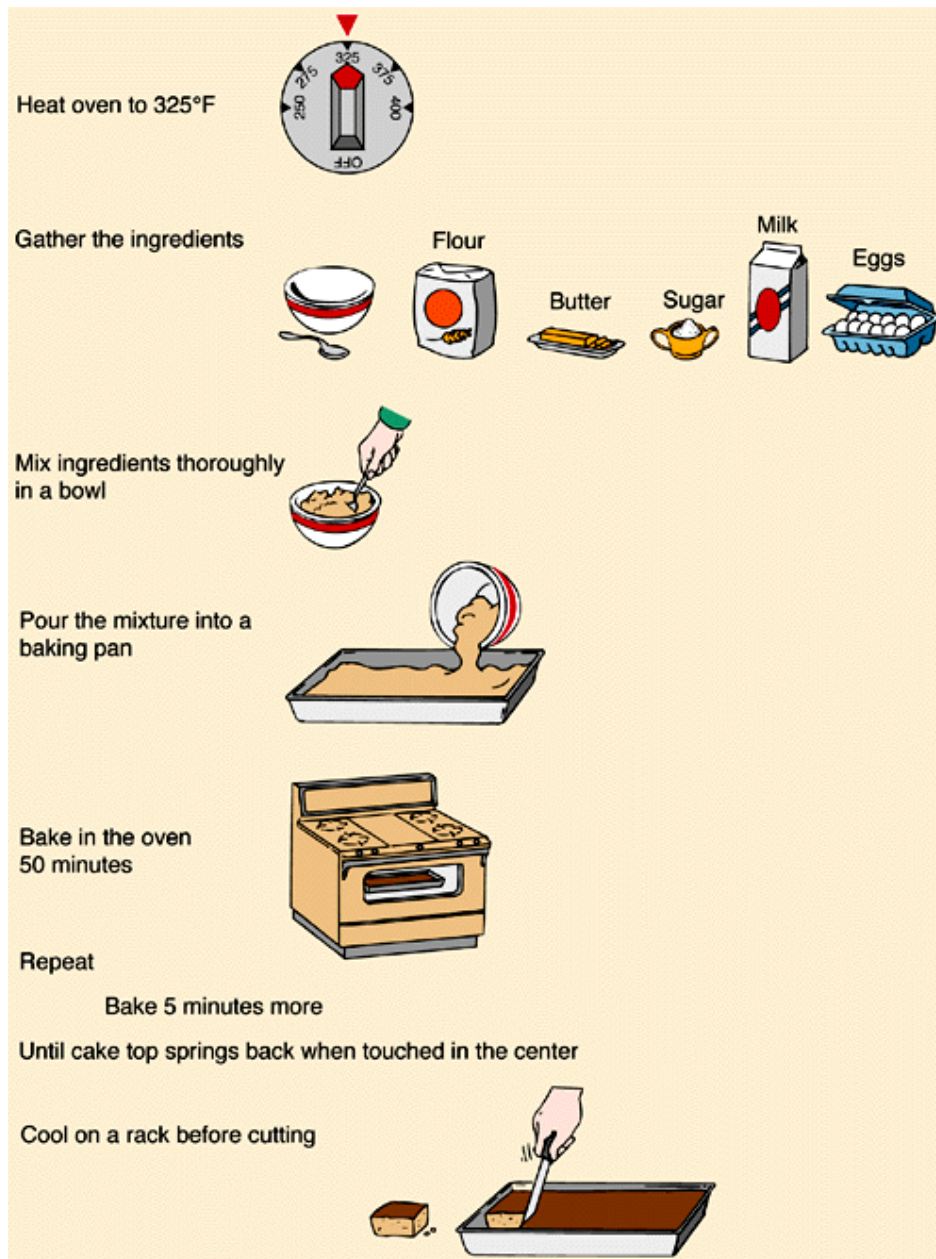
```
SELECT NAME, AVG(GRADE)
FROM STUDENT JOIN TERM1_GRADES
ON STUDENT.STUDENT_ID = TERM1_GRADE.STUDENT_ID
WHERE TERM1_GRADES.CLASS = 'ABC'
```

## Standard Query Language / SQL (High Level)

- In many programming languages, you are supposed to write the program as a procedure to do something

`al-go-rithm` : an ordered set of steps to solve a problem.

- Good algorithm should be:
  - Unambiguous
  - Have a defined set of input
  - Produce a defined set of output
  - Guaranteed to terminate
  - Guaranteed to produce a correct result



unambiguous

defined set of input

produce a defined output

always terminate

always produce correct result

# Testing and Debugging

Programming	writing procedures to achieve an objective
Testing	executing the program to check that the program can achieve the objective
Bug	problem in the program that prevent the program to achieve the objective in some or all situations
Debugging	fixing a bug in your program

# Programming Concepts

- Variables
  - Named storage for values
- Data Types
  - Different kind of values
- Expressions
  - Computation done on one or many values
- Conditionals
  - Provide a condition when the instruction should be performed
- Repetitions (loops)
  - Repeat a set of instructions until certain condition is met
- Function/Procedure/Method
  - A named set of instructions with a defined input and output

# Example – Bubble Sort

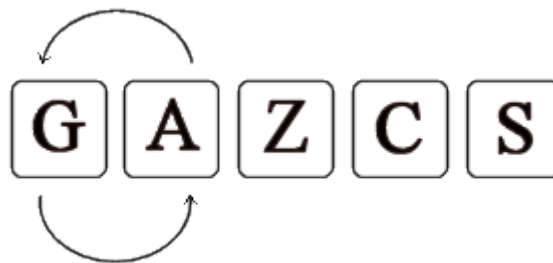
Suppose you have a row of children's toy blocks with letters on them. They are in random order and you wish to arrange them in alphabetical order from left to right.

Step 1. Begin with the first block. In this case, the letter G.



Step 2. Look at the block just to the right of it.

Step 3. If the block to the right should come before the block on the left, swap them so that they are in order

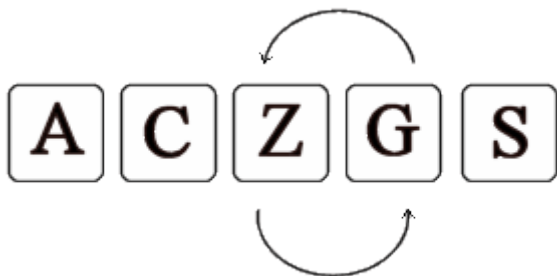
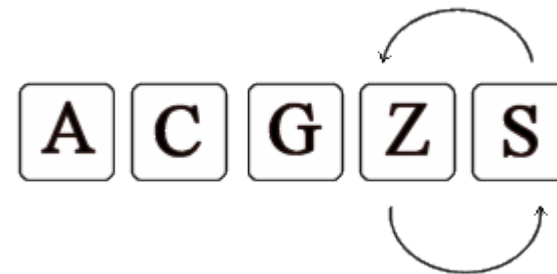
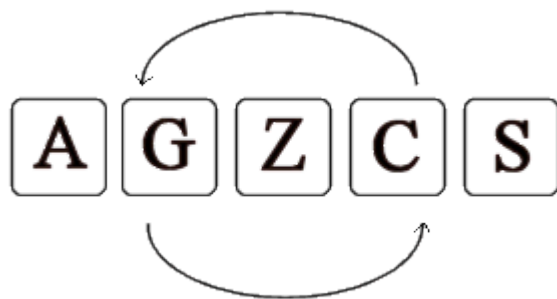




# Example – Bubble Sort

Step 4. Compare the next block in line with the first, and repeat step 3. Do this until you run out of blocks.

Step 5. Then begin step one again with the second block (next block of the block used in step 1). Repeat until there is no more next block.



# Example – Bubble Sort in C#

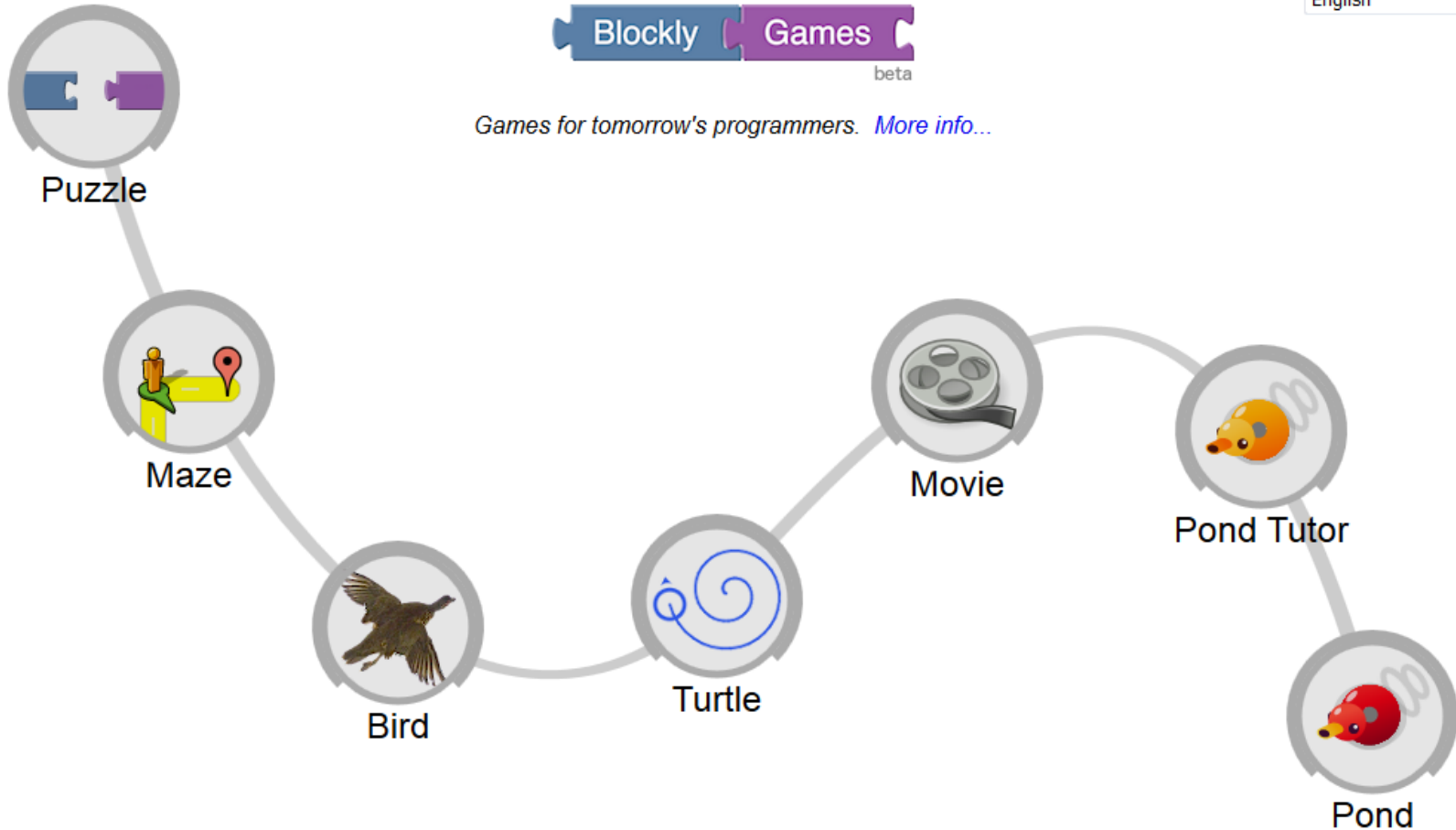
```
static void BubbleSort(Comparable[] array)
{
    int i = array.Length - 1;
    while(i > 0)
    {
        int swap = 0;
        for (int j = 0; j < i; j++)
        {
            if (array[j].CompareTo(array[j + 1]) > 0)
            {
                Comparable temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
                swap = j;
            }
        }
        i = swap;
    }
}
```

# First Step to Programming

English ▼



Games for tomorrow's programmers. [More info...](#)

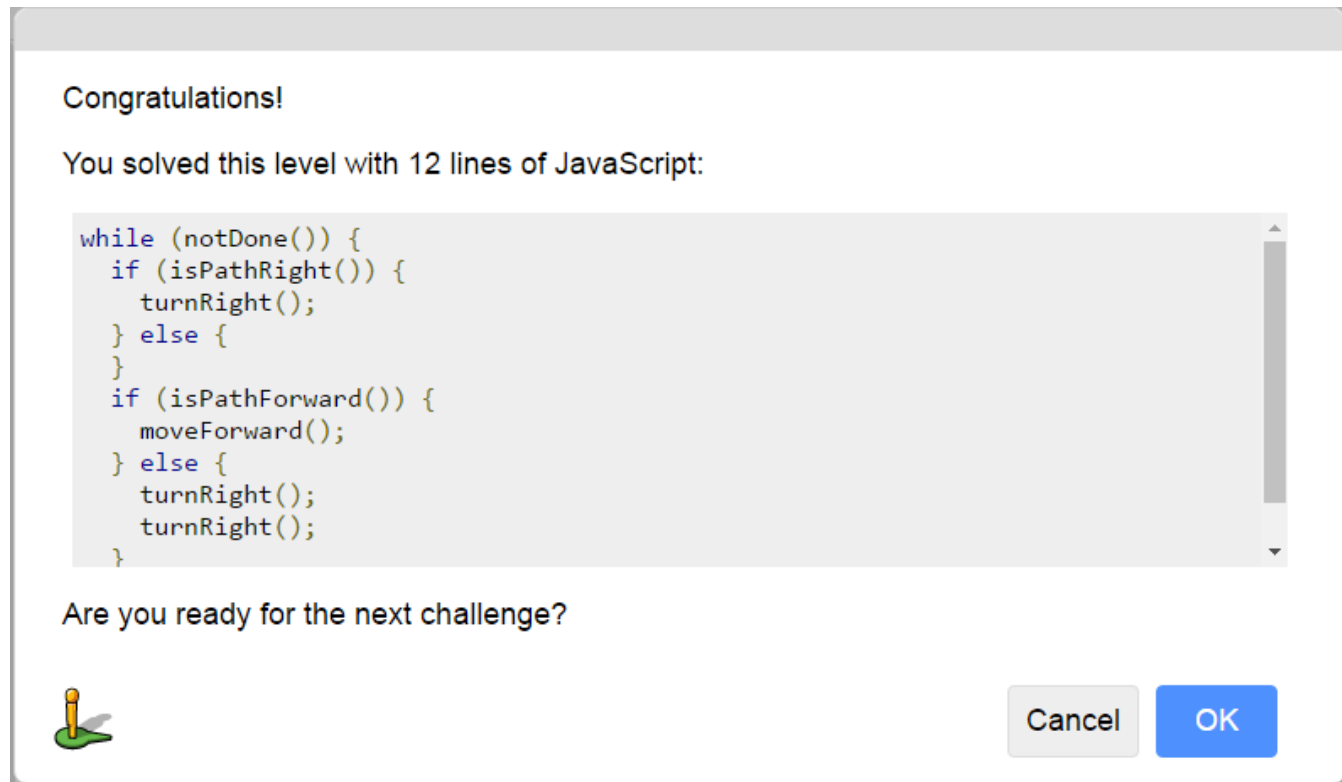


# Instructions

- Try the different games
- It is OK to get stuck in some level. Discuss with others to compare notes and ideas. It will help to cultivate your programming skills
- You may not be able to come up with general solution to the problem because you are only allowed to use limited number of capabilities.
- Don't give solution to others unless you're asked. Giving hint is probably better than direct solution – respect other people's rights to learn

# Suggestion

- Once you're done, copy and paste the JavaScript code to a Word document and keep it for your reference



# Expected skills learned

- Maze
  - Path finding
  - Repetition and condition
- Bird
  - Setting up a rule
- Turtle
  - Programmatic drawing
  - State management
- Movie
  - Keyframe animation
  - Variables and expression
- Pond
  - Autonomous agent like robots

# Summary

- What is programming?
- How do you program a computer software?
- Where does computer software located?

# References

- Blockly Games
  - <https://blockly-games.appspot.com/>
- Scratch - a drag and drop programming language
  - <https://scratch.mit.edu/>